

Illustration Watermarks for Vector Graphics

*Henry Sonnet, Tobias Isenberg, Jana Dittman,
Thomas Strothotte*

Department of Simulation and Graphics
School of Computer Science
Otto-von-Guericke University of Magdeburg

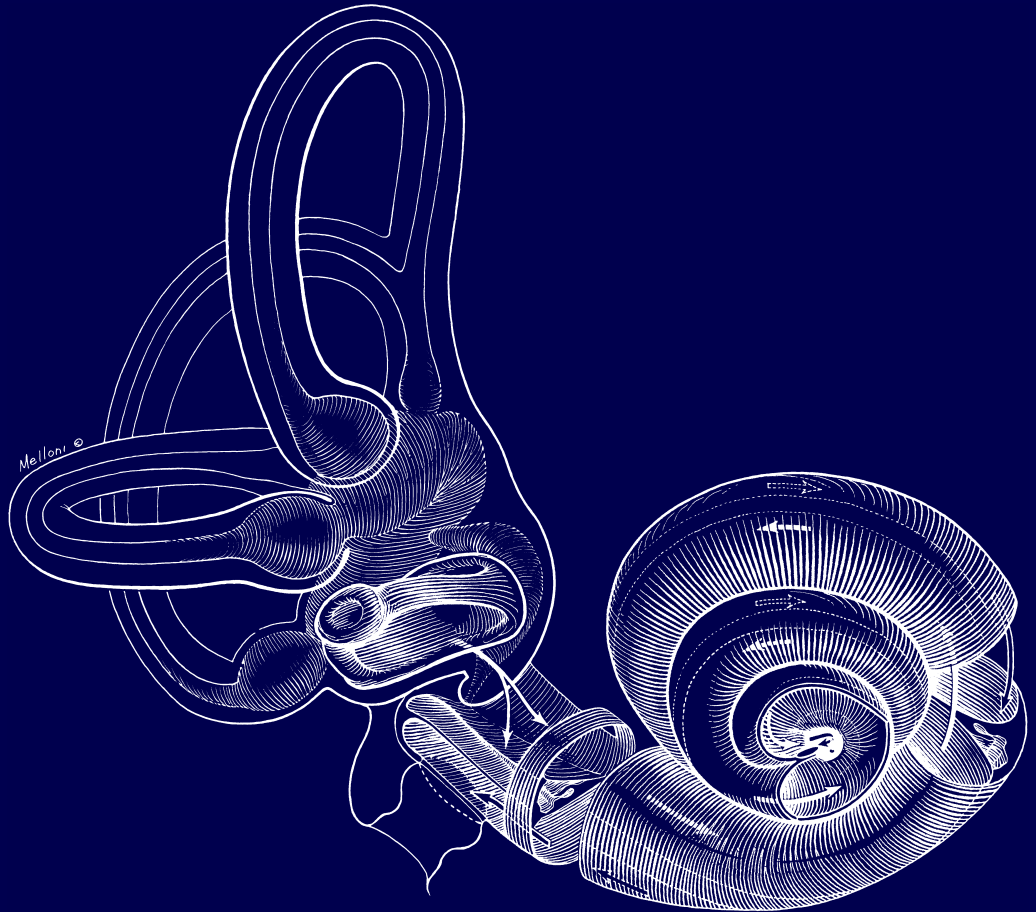


Outline

- Motivation
- Illustration Watermarks
- Process Overview
- Methods for Embedding Illustration Watermarks
- Properties and Results
- Conclusion
- Future Work

Motivation

- Illustrations
 - types
 - applications
- Line Drawings
 - abstraction
 - annotations
 - vector data
- Problem:
 - storing information about the illustration
 - information frequently lost in extra files



⇒ **Embedding the Information**

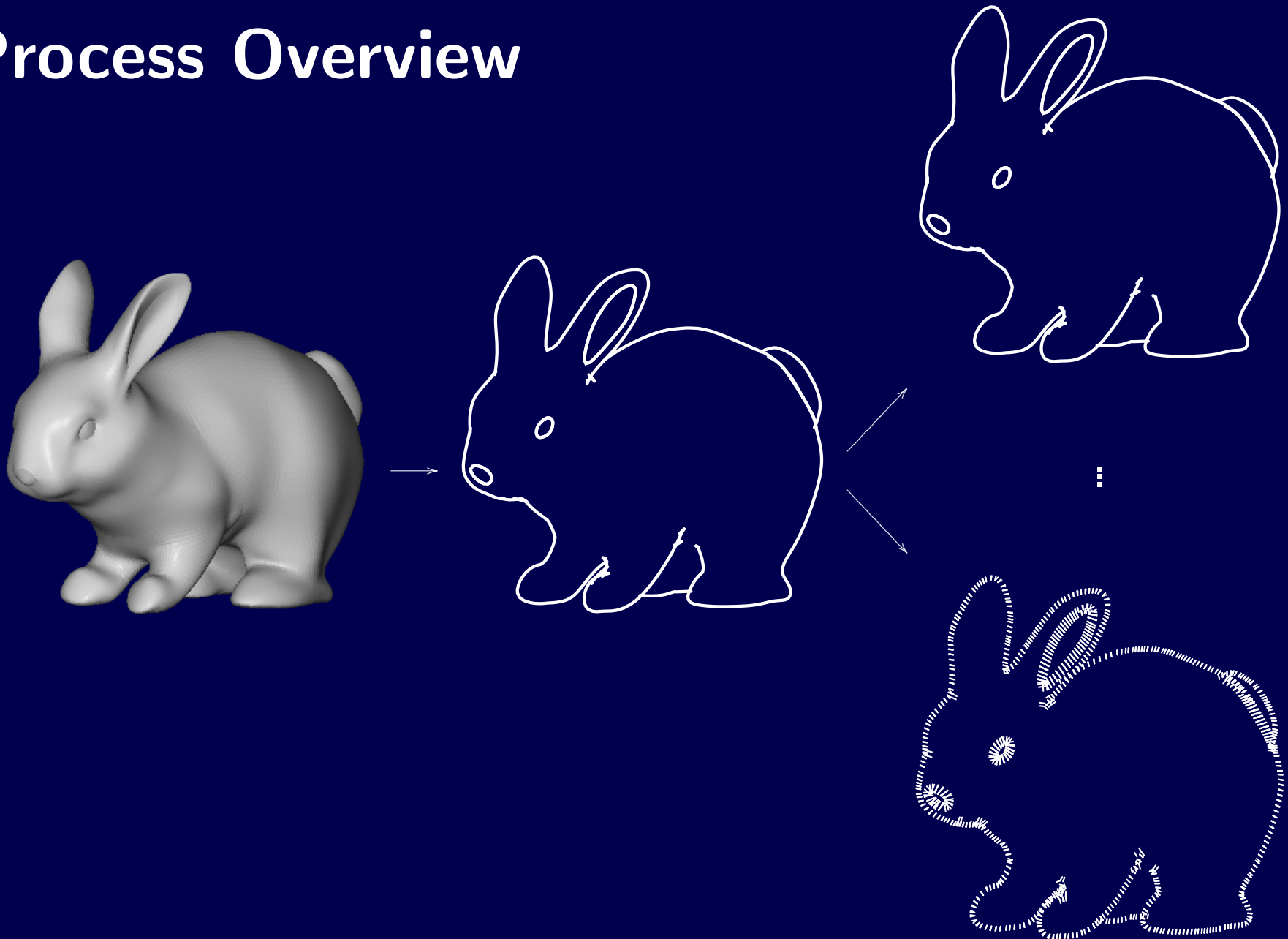
Illustration Watermarks

- Adding additional information to illustrations for
 - image enrichment
 - context dependent visualization
 - Examples:
 - annotations of objects that are shown on demand
 - additional illustration objects that can be blended in
 - rendition style (e. g., textures for lines)
 - contact information
- Goal is not to be absolutely secure
- Goal is to embed information with the vector data
- No changes to vector graphics formats necessary

Process Overview

- Silhouette renditions to generate vector images
- Use vector graphic parameters to embed data
 - stroke attributes (e. g., width, color, etc.)
 - stroke path
 - stylization
- Preparation for storing more data
 - stroke segment subdivision
 - compression algorithms for binary data
- Embedding binary data in the stroke sequence
- Storing in PDF format

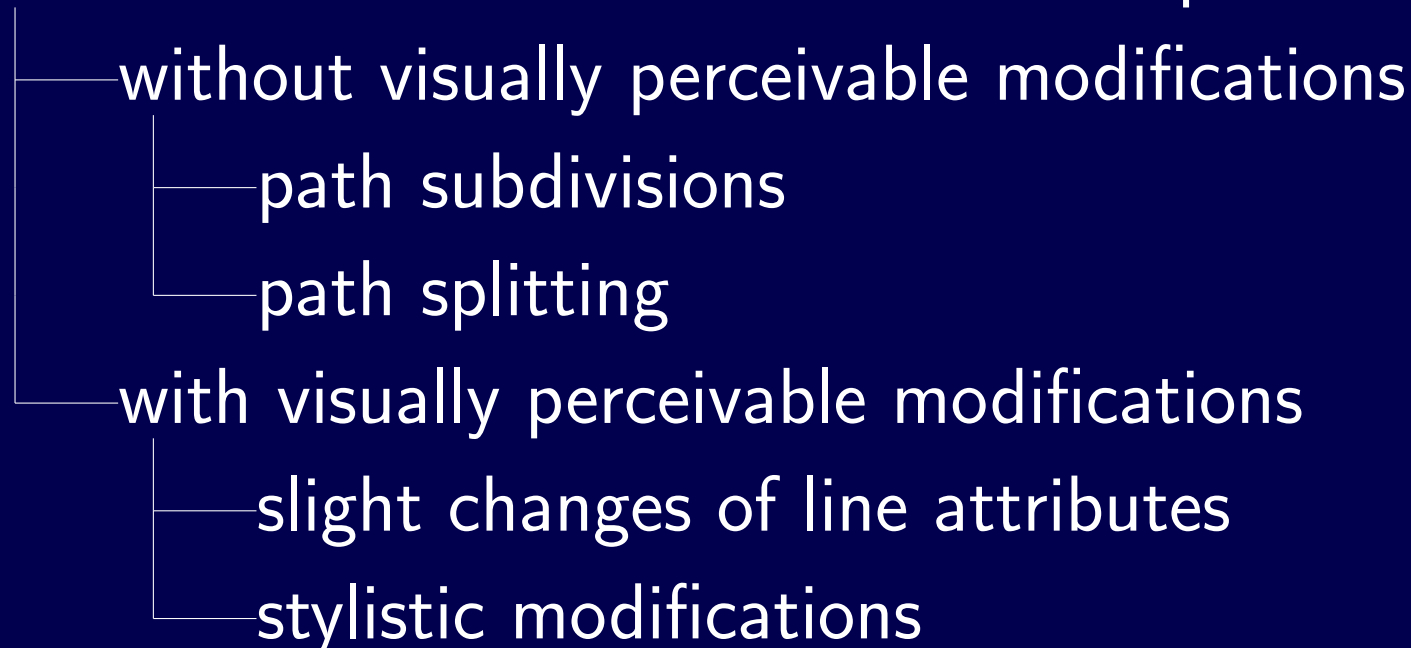
Process Overview



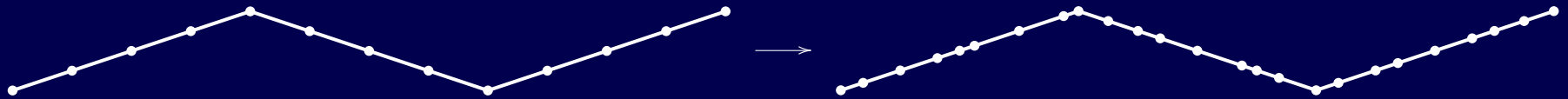
Classification of Methods

- Classification according to the degree of visually perceivable modification

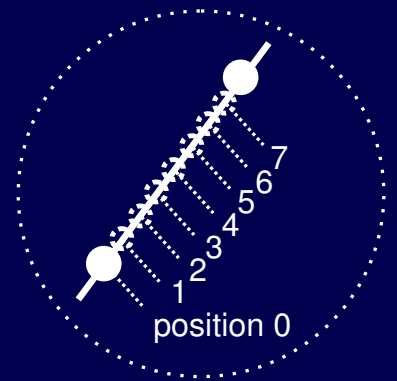
Illustration Watermarks for Vector Graphics



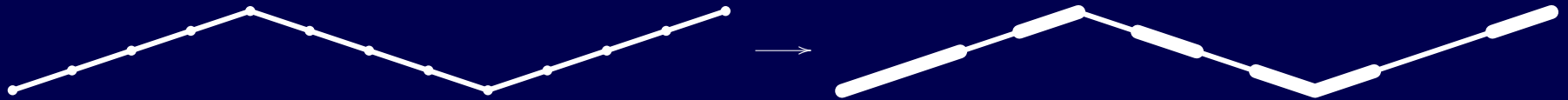
Path Subdivision Method



- Subdivide segment and add new vertex
- Encode bits by position of new vertex
- E. g., 8 positions = 3 bits per segment
- Alternative methods:
 - split paths and go back a bit for each segment
 - use line segment length (small base length)
- Visually not perceivable

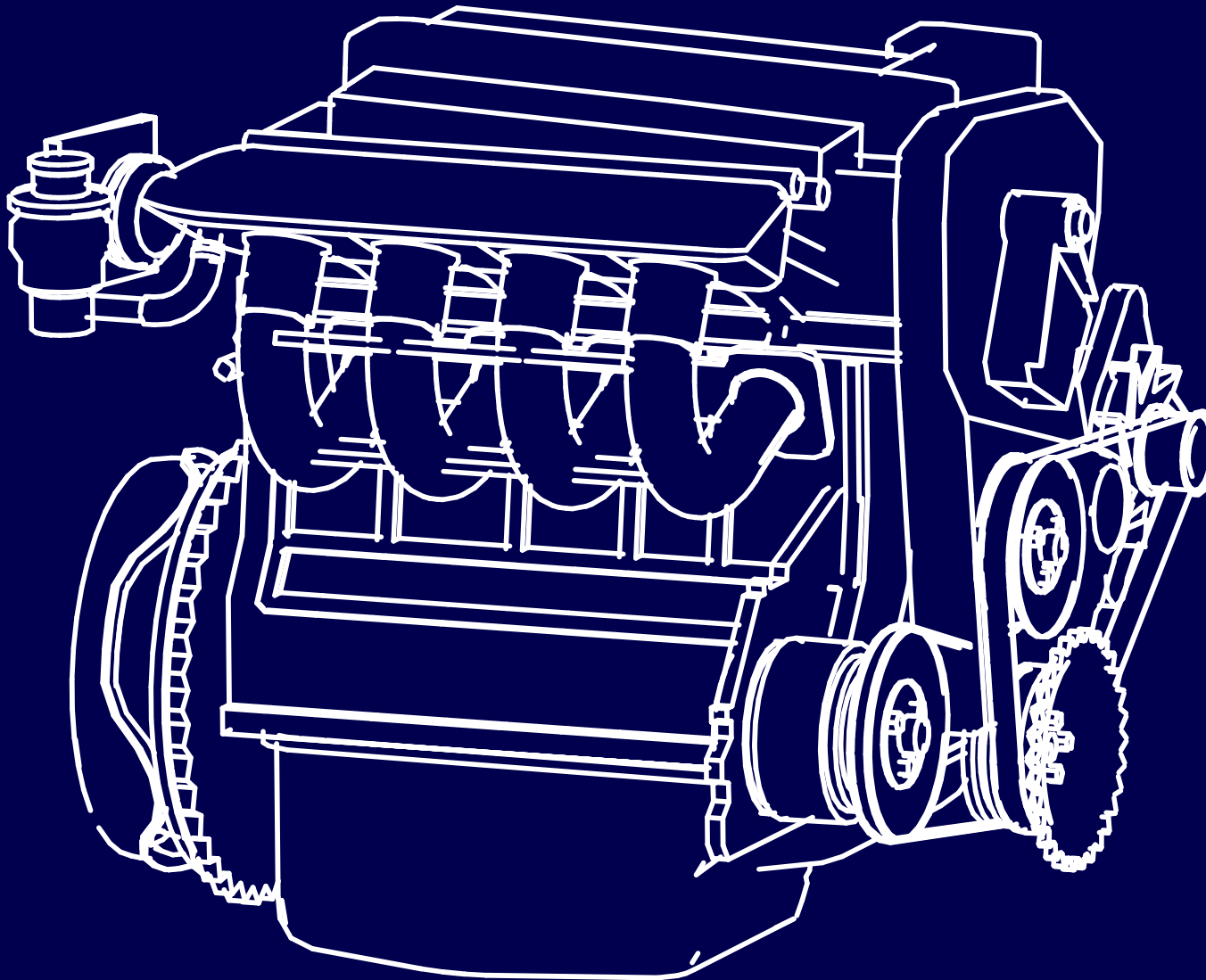


Line Attributes Method

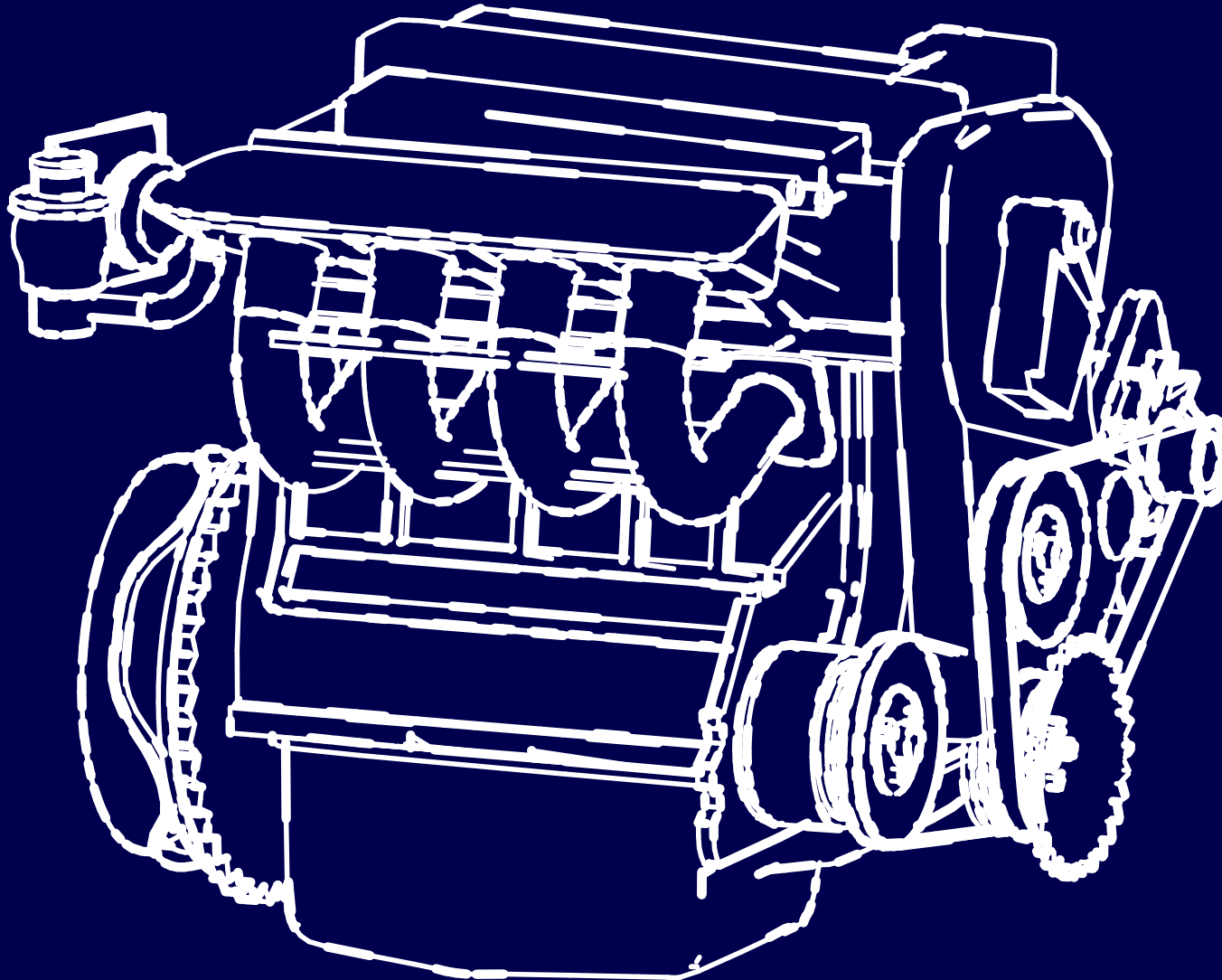


- Encode bits in slight attribute variations
- Example: line width
 - starting with a base width
 - increasing the width for encoding data
 - e. g., 4 width steps = 2 bits per segment
- Other attribute: color (problem with halftoning)
- Almost not visually perceivable when steps are small
- Often means splitting the path (e. g., in PDF)

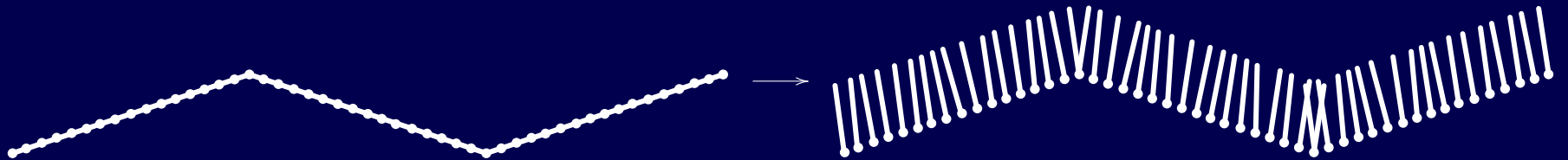
Line Attributes Method



Line Attributes Method



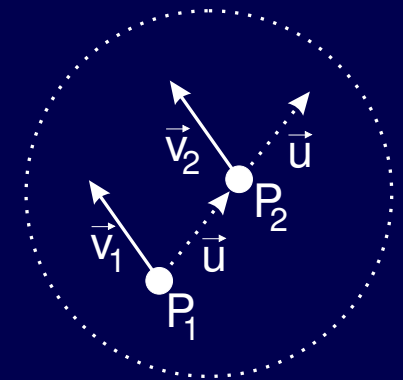
Angled Lines Method



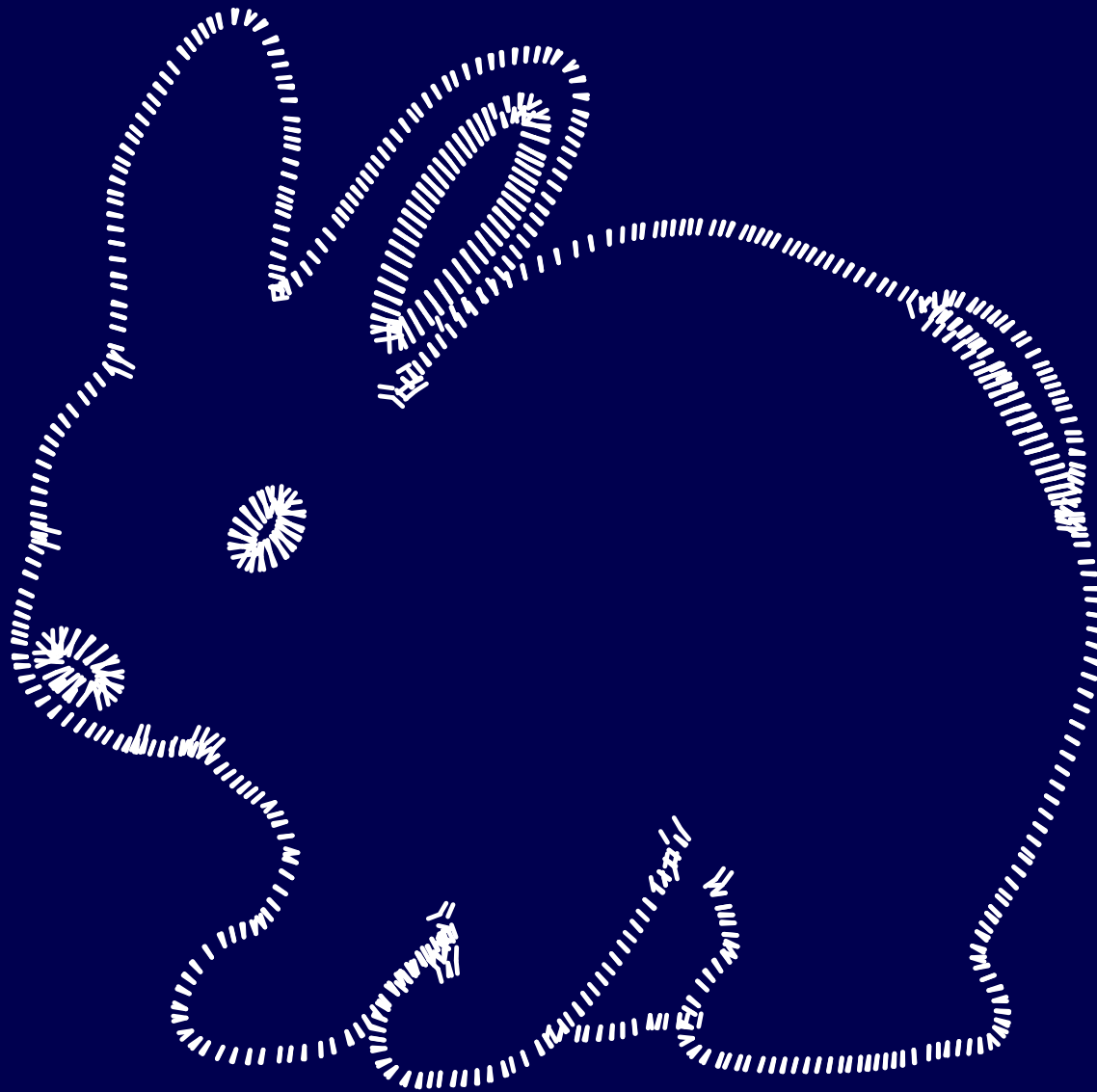
- Encode the bits using the angles between strokes
 - first stroke per original segment as reference:

$$\beta = \angle(\overrightarrow{v_1}, \overrightarrow{u})$$
 - angle with offset t encodes data:

$$\beta' = \angle(\overrightarrow{v_i}, \overrightarrow{u}) = \beta + t$$
 - e. g., 8 possible $t = 3$ bits per stroke
 - high degree of segment subdivision needed
 - new reference when \overrightarrow{u} changes
- Artistic effect



Angled Lines Method



Properties of the Methods and Results

- Perceivability of changes: 3 degrees
 - no visible changes
 - changes not perceivable (on screen and in print)
 - stylistic appearance
- Robustness
 - processing in vector graphics tools
 - common transformations (in particular scaling and rotating)
 - non-linear transformations (works for line attributes)
- Partial deleting: part of message still readable
 - object-dependent coding possible

Properties of the Methods and Results

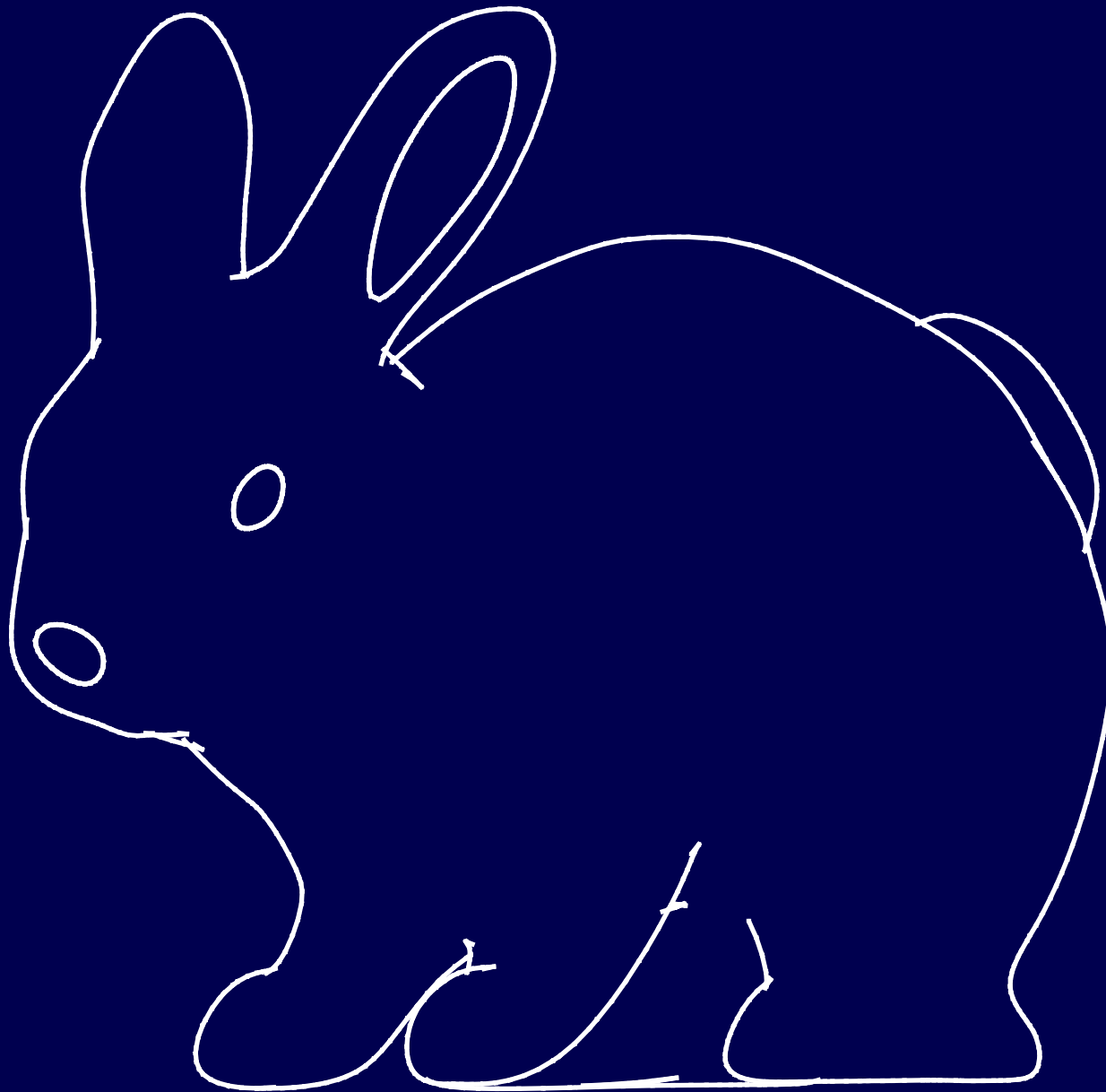
- Capacity
 - additional line attributes expensive
 - line splitting expensive
 - approximately linear behavior depending on size of original file
- Capacity improvements:
 - regular compression of binary data
 - binary PDF coding
- How easy to detect and to remove?
 - additional line attributes easily removable
 - path subdivision easiest to handle but easy to remove
 - split paths easily detectable and harder to remove

Conclusion

- Illustration watermarks
 - embedding data with illustrations instead of separately
 - vector graphics
- Various methods, most without perceivable changes
- Robustness to common vector graphic processing

Future Work

- Come up with and examine more methods
- Embed into illustration application
- Interaction possibilities



Capacity

